



**SCHOOL OF COMPUTER SCIENCE**

**ANU COLLEGE OF ENGINEERING & COMPUTER SCIENCE**

## **3D Visualization of the Curriculum**

Student: **Jian Guo (U4809721)**

Supervisor: **Lynette Johns-Boast**

CourseCode: **COMP8780 - Human Centered Computer Project**

**June 3<sup>rd</sup>, 2011**

# Abstract

The complexity of the structure in curriculums often makes some education programs even the standard degree program offered by a university cannot reach the intended outcome as they are designed to. In this report the author suggests a new approach-by using the 3D visualization to demonstrate the relationships of the program- to help the course designer to make the structure of the program more reasonable. This report mainly focuses on the method to visualize the information and internal relationships among the curriculums in a 3D graph. The theoretical basis of this project is the Constructive Alignment theory advanced by Biggs [1].

After using Constructive Alignment Theory to building the data model, the author applies the Java3D technique to build the scene graph which demonstrates the information in a 3D graph. The result shows that the 3D visualization technology really helps demonstrate the relationship and help understand the complexities in the program. The Data based design can support the visualization process very well. However the Java3D lack some support in interaction with the object in the graph. Also, there are some desired features which are not implemented such as manipulation of the objects in the Graph (delete, edit, add, drag) and these features will be achieved in the future work.

## 目录

Abstract.....	II
1. Introduction .....	1
1.1 Objectives.....	1
1.2 Structure of the report .....	1
2. Background .....	2
2.1 Constructive Alignment .....	2
2.2 Visualization.....	4
2.2.1 Why visualization.....	4
2.2.2 Techniques of visualization on web .....	4
2.3 B-S model .....	6
3. Motivation.....	6
4. Methodology.....	7
4.1 Statement of the requirements of the project.....	7
4.2 Research on A-content .....	8
4.3 The step to do the visualization.....	9
4.4 The structure of the program: .....	15
4.4.1 MVC design model .....	15
4.4.2 The architecture of the program .....	15
4.4.3 The design of the classes .....	16
4.5 The result of the Visualization .....	22
4.6 Test .....	23
4.7 Problems and solutions.....	26
4.8 Analysis of the result.....	27
4.9 Reflection .....	28
5. Future work.....	28
6. Conclusion.....	29
7. References .....	31
8. Appendix.....	32

# 1. Introduction

## 1.1 Objectives

With the development of the education industry, the degree program in universities become more and more complicated, and to construct a program of high quality is a urgent requirements of the university as well as the student themselves. However a degree program curriculum is a complex network interlinking learning requirements the aim of which is to eventually produce a graduate who would exhibit the attributes expected of a graduate of that specific program.

So in this situation, we must find a criterion to instruct the course designer how to arrange the courses properly and efficiently. The Constructive Alignment [1] theory argues that learning activities and assessment tasks should be aligned with the intended learning outcomes of a course. In this way we can enhance the clarity of the curriculum design and provide transparency of the links between learning and assessment leading to improved learning outcomes for students.

To design the whole program, we have to check the relationships among the intents, aims, goals, objectives, contents and the other complex relationships among the curriculum of a specific program. However, a degree program consists of multiple courses, all of which have their own set of intended learning outcomes (ILO). This usually is very complex and confusing. So we have to clarify the complexity with the help of 3D Visualization techniques. With the help of this project the course designers can easily ensure that all the courses in a program together have appropriate learning and assessment activities and this ensure that a program's intended outcomes can be achieved. Thus, my job is to do a research on how to use 3D visualization techniques to show visualize the relationships for the whole program.

## 1.2 Structure of the report

The rest of the report is structured in the following parts:

- Part 2: Introduce the background of the project. Including the introduction of theory basis- Constructive Alignment, B-S model, the term Visualization, and why we apply visualization on the data. Furthermore, I listed some options for

implementing information 3D visualization.

- Part3: Introduce the motivation of doing the project. How this project came about, what I want to achieve by doing this project.
- Part 4: methodology, this is the main part of this report. In this part, I state the requirements of the project, the research done on the A-Content. After that, I introduce my own method of visualize the curriculum information. Then I summarize the structure of the program and give detailed explanation on why I choose to construct the class like this. After I show the result of the visualization, I test the application using verification test method. Finally, I make a summary about the problem solutions and reflection.
- Part5: in this part I state some features which is really attractive however they are not implemented in this stage. All of these features will be left as future work.
- Part6 Conclusion: In this part, I make a conclusion of this project, stating what I achieve what is not achieved, and why some thing is not achieved.

## 2. Background

### 2.1 Constructive Alignment

This project is based on the theory of constructive alignment [1], which gives me the idea of reveal the relationship between ILOs, learning/teaching activities and assessment methods which will be critical for whether the course is aligned.

Introduction of constructive alignment

Constructive alignment is a theory worked out by Biggs which focuses on making the learning activities and assessment tasks aligned with the learning outcomes that are intended in the course. “The constructive alignment theory is rooted in curriculum theory and constructivism”[2]. The relationship between ILOs, learning/teaching activities and assessment methods can be shown in figure1.1 [3]

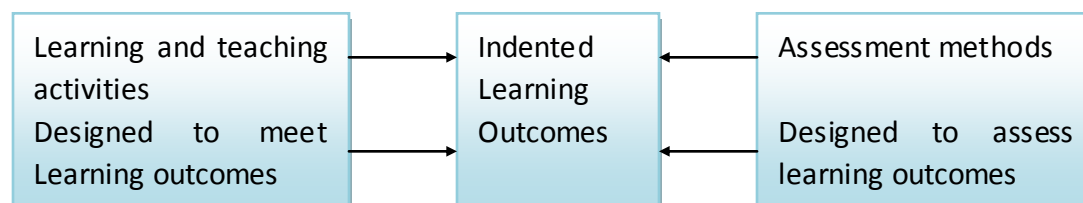


Figure 2.1 Relationships between ILO, activities and assessment

And there are three aspects that are important for teacher. The first is to encourage and support the activities which are helpful to achieve the ILOs. The second is clarifying what we want the student to be able to do at the end of the course and

make it clear to the students themselves. The last thing is to design the assessment method carefully which really reflect the learning outcomes we want students to achieve.

Furthermore we have to adjust the program structure regularly. In the teaching and learning process, we usually encountered some unexpected learning outcomes that are useful to students. In this case, we should modify the original ILOs to include the new learning outcomes into them. In the other aspect, for those courses content which don't contribute to the ILOs, we should delete them.

Figure 2.2 is adapted from the web site [3] which can be a display of the adjustment of courses

There are 4 criterions for constructively aligned course [4]

- The intended learning outcomes (ILOs) are stated clearly;
- The ILOs are explicitly communicated to the students;
- The exam assessment(s) match the ILOs;
- The teaching form(s) match the ILOs.

To sum up, the Biggs's theory can be a good guide for constructing course and programs.

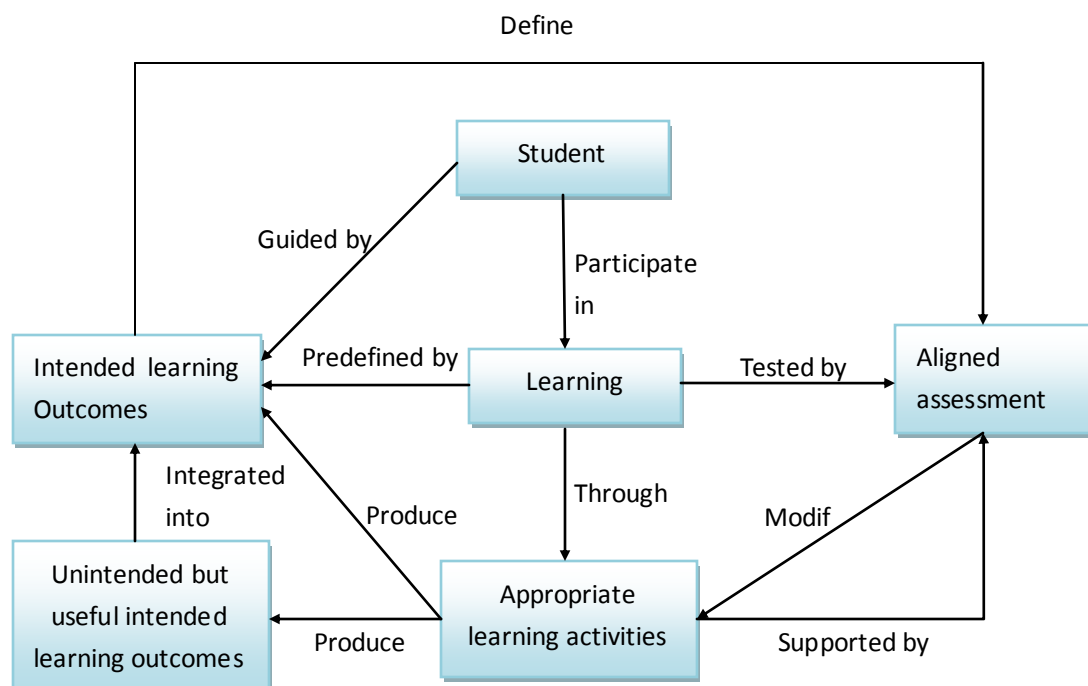


Figure 2.2 The procedure to adjust a course

## **2.2 Visualization**

There are a few terminologies to describe visualization: Scientific visualization using in the computer science area [5] Data visualization which is more general than scientific visualization, and is used to interpret data source from finance or market [6] and information visualization which endeavors to the representation of large-scale collections of non-numerical information library, networks of relations on the internet, and so forth. [7]

The main difference between information and scientific visualization is the dimension that is used to represent the data, the latter contains normally up to 3 dimensions and the former is suitable for higher dimensions or abstract spaces. In my project of visualizing the relationships among curriculums, a relationship in a specific course and program is usually more than three dimensions and sometimes, the dimension is abstract, so we will apply the technology of information visualization.

### **2.2.1 Why visualization**

Advances in computer hardware have made the specialized 3D graphic hardware very common even on the ubiquitous personal computers. 3D images, visualization is more and more important in the computer graphic area. On the other hand the rapid development of software, hardware as well as the network, information in another word data is brimming over, without visualization, we can hardly make use of such a huge amount of data. By represent the data into graph which is more comprehensible for human because in this way we can make good use of our highly developed vision system.

To sum up, we can expect to grasp some information hidden under the data by using visualization techniques.

### **2.2.2 Techniques of visualization on web**

Web visualization has the advantage of sharing the information all over the world quickly and efficiently. Web visualization shares most of the features which computer visualization has to communicate large amount of data. So visualization by web is now very important and is becoming more and more widespread.

### ➤ **Early web Visualization**

The early Internet language is HTML, which defines the content and image in a simple document which can be transmitted over the internet. However at that time images on the web are static; in another word predefined; meaning the user can not interact with it. Later a new technique called Common Gate way Interface (CGI) along with HTML enables the client to view a lot of dynamic graphs, including animations, which can be controlled and interacted with by the users. The client sent information to the server with the help of a HTML form. Then the server executes a CGI-BIN script which can generate a new MPEG picture according to the user's request and responds the final result to the client. However this technique put nearly all the work load to the server which may use up the resources on the server.

### ➤ **Virtual Reality Modeling Language (VRML)**

The Virtual Reality Modeling Language (VRML) is used to construct real world or mythical 3D model over the web which contains a scene graph representing a 3D world. Furthermore, this language enables the user interact with the scene and the scene can change according to the action of the user. Now VRML is becoming new international X3D standard which is a 3D technique which supports XML language and has the feature of portability and page integration.

### ➤ **Java 3D**

Java 3D is an extension of Java language in the field of three dimensional graphics and it is an Application Programming Interface (API). JAVA 3D provides the development of three-dimensional entities to create, manipulate and color from high level , so that development work has become extremely simple

Java 3D provide the creation, manipulation, and coloring of three-dimensional entities on the high level, which let the programmer focus on the development of 3D object and 3D virtual reality scene not the low level rendering details. This great reduce workloads of development work. Meanwhile, the lower level of Java 3D is based on Direct 3D, OpenGL and so on.

Java 3D has the following features:

- Good portability, cross-platform features.
- Support most of the graphic format, 3DsMax, OBJ, VRML, X3D and so on, which means we can import the model from the other software and then integrate into the program.
- Java 3D support the latest virtual reality sensor and display tools.



- Java 3D supports the spatial sound effect.
- Java 3D have a collision detection function.
- Java 3D is a set of API based on Java 2, so, it easy to integrate with Java which can handle a lot of complicated problem and have a rich library.

#### ➤ **Final choice made on these techniques**

Base on the analysis of the main techniques, we choose Java3D because it can fulfill nearly all the 3D scene creation task as VRML.

It is an object oriented language, so we can view the 3D object as objects and assemble the 3D objects together to form a much complicated scene. Furthermore, by using this feature, we can make our 3D object reusable, and provide a perfect interface for later development.

Java 3D is rooted in Java2, which can easily integrated with Java, and the rich resource in Java can be an advantage.

## **2.3 B-S model**

B-S model refers to the browser and server model, which divides an application into two parts, one is the Browser side and the other is the server side and the two parts work together to provide the result or the browser side the user simply requiring a web browser and then sends requests to the server. After receiving respond from the server the browser interprets the http code and renders the information of the http response into web pages. In this way the user can easily get the information required without installing other software. So, it is easy to maintain and upgrade.

In the C-S model, the client have to install a client application on the local computer, which will in charge of some computation whereas in the B-S model the server always do a lot of job, and it is easy to break down due to the large amount of request from the client. So, we should deal with the trade off and balance the load between server and client side.

## **3. Motivation**

According the Biggs's constructive alignment theory for courses, the quality of the whole program in a college can be greatly improved.

Based on the literature review over previous research, there is not much visualization application made to help alignment the courses. So I conduct a project which can be used to demonstrate the relationship of the curriculum that has already stored in the database automatically. The relationships are simulated by a 3D graph, which can easily show the complicated spatial structures.

By viewing this graph of the relationships, the user can easily find the faults or drawbacks for the construction of the program. For example, you can find which Intended Learning out comes is not well supported, which content or delivery method can be deleted without breaking the whole structure of the program. What's more the user can see the detailed information for every ILO, aims and so on in a program.

Meanwhile, the result of this project can provide some advices on how to organize the graph to demonstrate the relationship in the program.

## **4. Methodology**

### **4.1 Statement of the requirements of the project**

1. The system should establish connection with A-Content and retrieve information from it.
2. Establish a database which is used to hold all the information about the programs, courses and delivery method also the relationship among them.
3. Create a high level view of the while program system.
4. Create a detailed level of a specific program, course and delivery method.
5. Integrate the 2 levels together.
6. Generate the relationship graph automatically from the database.
7. Manipulate the items in the graph to delete a node, add a node, change the information of a node, and view the information about the node.
8. Give suggestion about the each manipulation, for example if you delete a node the system will give you feed back such as, "if you delete the course, the ILOs will not be meet any more".

## 4.2 Research on A-content

This project is initially based on a web application named A-Content, which is a web based content authoring tool, which is used to construct courses and programs.

The first task is to set up connection with A-Content and gain information of the course from the server by http request.

By Set up an A-Content application on the local sever at port 8080, I test the web service API offered by the A-Content. The procedure can be listed as follows.

First Download the Easy-PHP and set up it locally, which contains the PHP, apache server and the MySQL database server, and the configuration is automatically generated which may help avoiding making mistake by set the configuration manually.

Then, write a class which can set up establish with the server, then using the http request to communicate with the server.

We can access the A-Content by the Web API offered by the A-Content [8]

➤ Search on the A-Content API:

The URL address used to Search the A-Content is:

<http://atutor.ca/acontent/demo/search.php?id=fb86&keywords=coursename>

This means the ID for the current user is fb86 and search for course name, and the result is formed in XML format, giving the information about course ID, course title, description about the course. The search result only contains the basic information if the course.

The URL to View the course is:

[http://atutor.ca/acontent/demo/home/index.php?course\\_id=2](http://atutor.ca/acontent/demo/home/index.php?course_id=2)

Here the course\_id is the result returned by the search above.

And the result is the HTML code of the first page of that course, which is meaningless and cannot be comprehended very easy. So I hold the opinion that this API is not useful in viewing the content or information of the course.

Luckily the API offers another way to download the course and the URL request can be listed as follows.

[http://atutor.ca/acontent/demo/home/imscc/ims\\_export.php?course\\_id=10](http://atutor.ca/acontent/demo/home/imscc/ims_export.php?course_id=10)

Here the `course_id` has the same meaning as view content.

And the server will return a ZIP common cartridge of that course and this common cartridge follow the standard of IMS standard, so we can do a research in the standard and then write an interpreter to get information from the common cartridge.

We can see that the exposed API is not detailed enough to retrieve all the information needed to construct the visualization.

However I would like to remain the web feature of A-Content even though we do not have a solution to gain enough information from the sever due to time limit and the job of connecting the project with A-Content remained to be future work.

So, to be consistent, with the A-Content which is a web application, I must choose the visualization techniques based on the Web technologies.

### 4.3 The step to do the visualization

From the figure 4.1 we can see that in the first step data is gained from some data source. Next, the data is classified and stored in a database and then I do some transformation of the data to make it more sensible. After that, map the data to an appropriate way to represent to the reader. Finally the data is displayed.

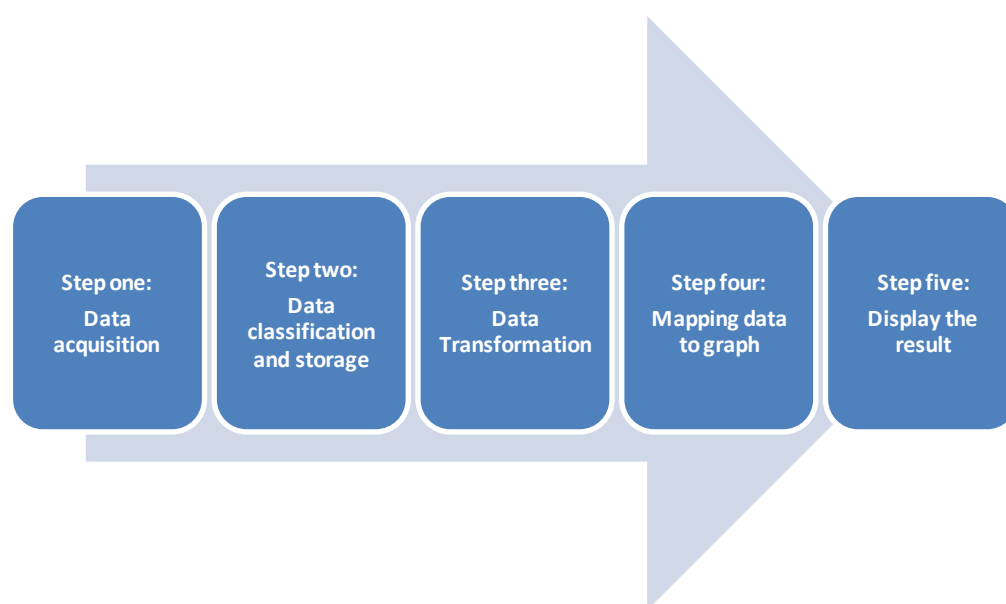


Figure 4.1 The step of Visualization

- The first step: Data acquisition

Data acquisition is the step of get the related course and program information from the school. The data source could be the school program information data base, or the school archives. And the format of data can be well structured data or documents.

- The second step: Data classification and storage

Data classification:

For the curriculum information, base on the unpublished paper of Lynette Johns-Boast I divide them it into several categories [9]

- program level information:

- Intent is the attributes we want students to have when they finish the whole curriculum. For example, the graduate students will have the essential skills in computer sciences area, and can be the leader for the future academic area.
- Aims are what abilities we want the student to display at the end of a certain program. For example, Graduate will demonstrate the understanding of software development process very well.
- Goals are more specific and are derived from the aims. For example, Students will develop skills such requirement acquisition, architecture design, applying design pattern into programming.
- Objectives are specific statements of program intent, which can be a guide for the course designer to arrange the content of a specific course or a program”.
- Body of knowledge is the domain areas that covered by a specific program.

- Course level information

- Learning activities: the actual activities we are doing during the learning process, such as the taking the lectures, attending the tutorials, going to labs, and doing assignments.
- Assessments / Tests: the teaching activities used to evaluate the mastery of a particular content of the course.

- Content: the core knowledge of a course and it is what actually contained in a course, the materials the lecturer imparted to the student.
- ILOs: Intend learning out comes.

➤ Delivery method level

- Delivery method: is by which way we present the course content to the students, e.g. audio, video, slides and so on.

Data storage:

Before storing the information into the data base, I need to design the tables that could contain the information above.

The E-R diagram shown which is quoted from the unpublished paper from Lynette Johns-Boast [9]

The high level design is shown in Figure 4.2

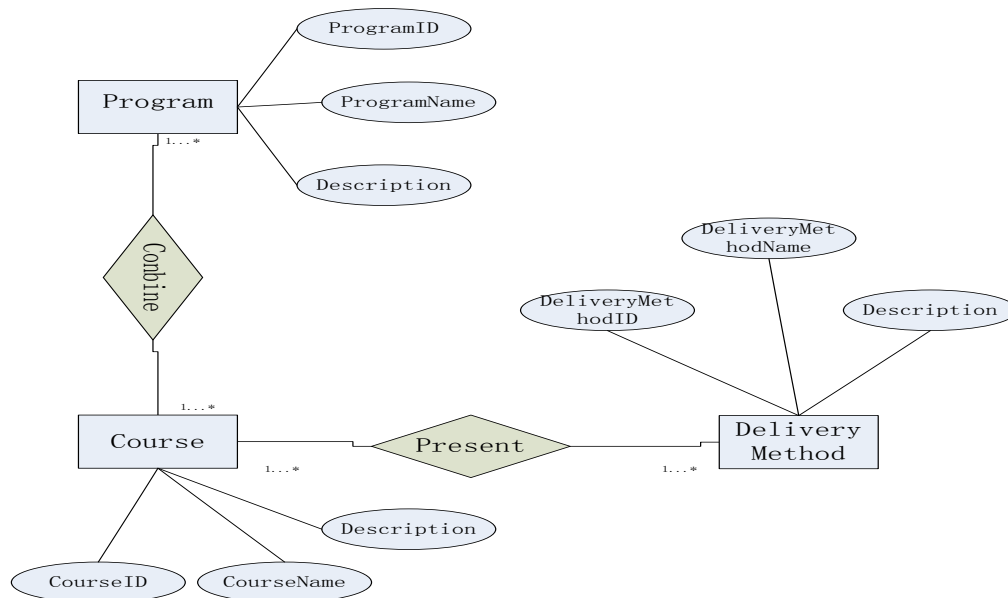


Figure 4.2 High level E-R diagram

The program level design is showed in the Figure 4.3

The course level design is showed in the Figure 4.4

The layer between program and course is showed in Figure 4.5

The layer between course and delivery methods is showed in Figure 4.6

- The forth step: Data transformation:

In this step, my main task is how to transform the information stored in the data base into a data structure which is easy to be visualized, which I will demonstrate later.

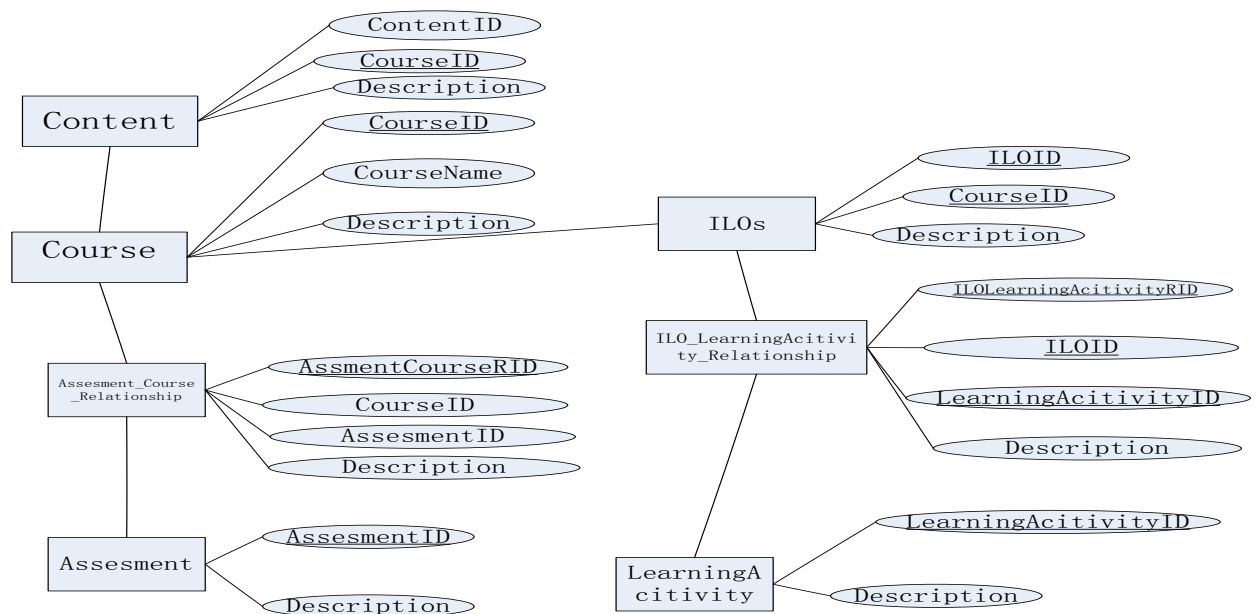


Figure 4.3 Program Level E-R diagram

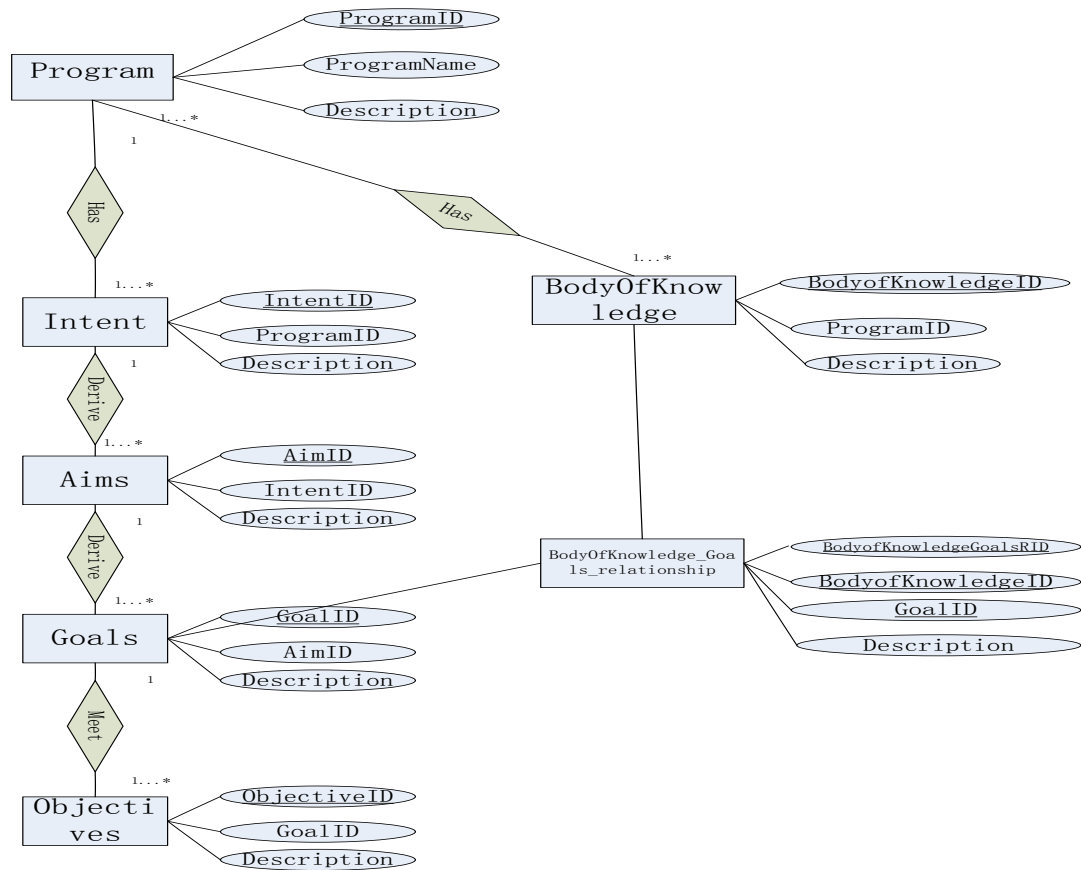


Figure 4.4 Course Level E-R diagram

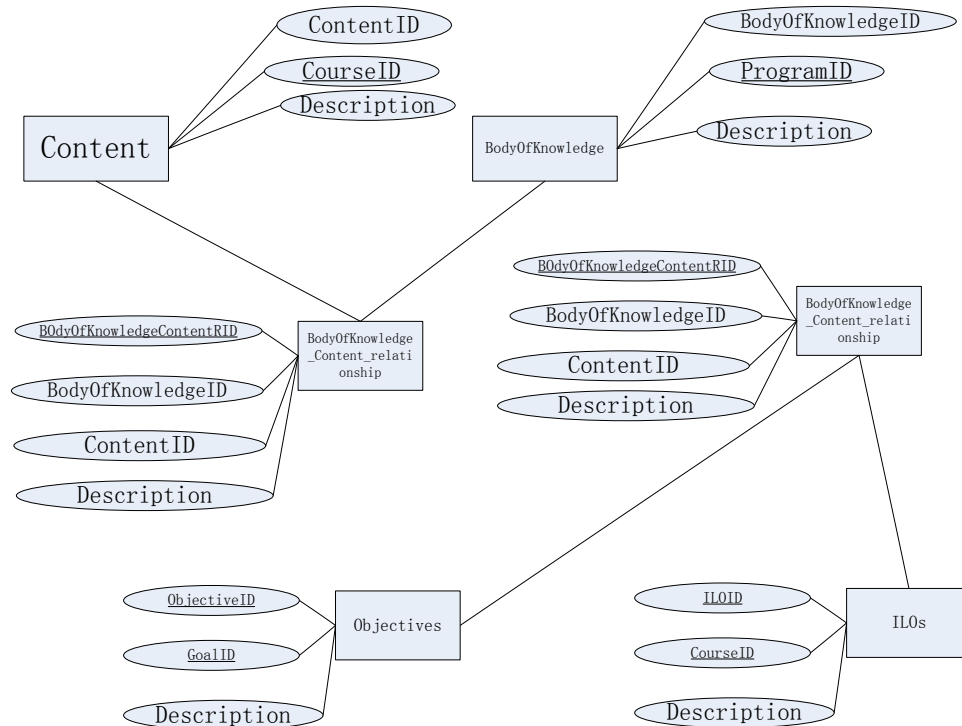


Figure 4.5 E-R diagram between course level and program level



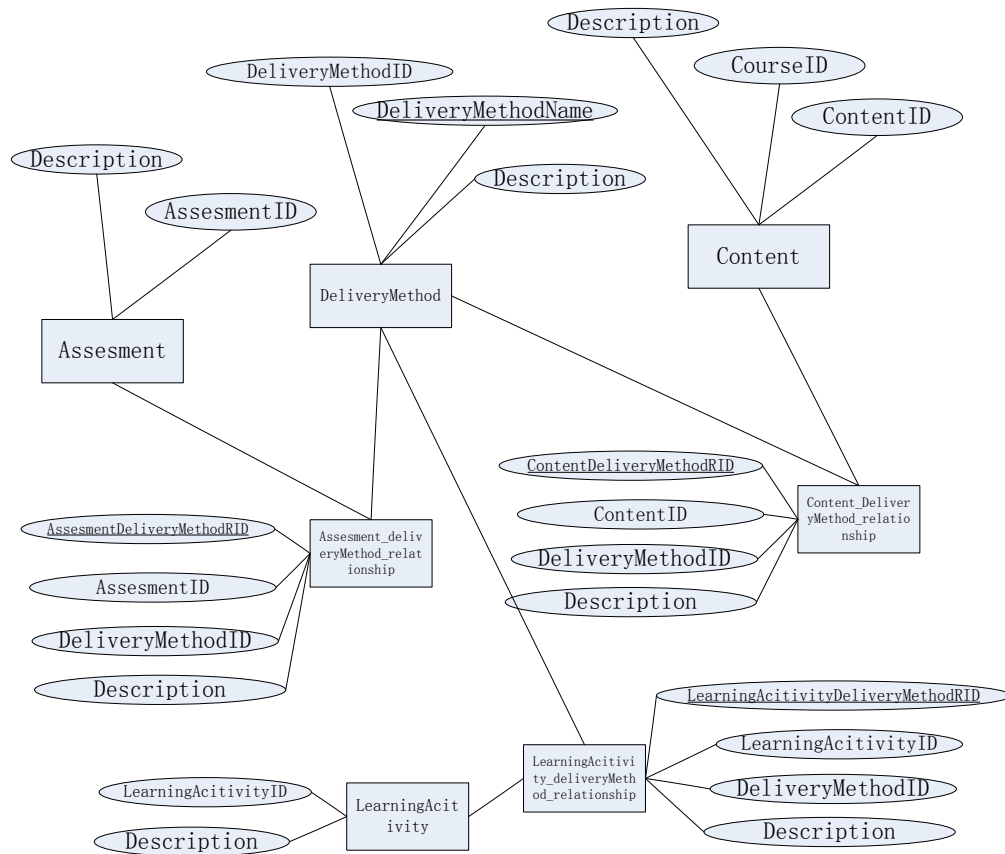


Figure 4.6 E-R diagram between course level and deliveryMethod level

- The fifth step: Mapping the data to graph.

The first question to consider is how we arrange the graph.

Based on the analyses before, our problem can be view at different level, for the high level overview, the graph focus on the relationships among program, course and delivery methods. So, in this graph, we show the programs, courses, delivery methods, and link them.

For the second layer, I go a step further into the detailed information in the program and course itself. For the program, we demonstrate the intent, aims, goals, objectives, body of knowledge, and link the relationship among them.

- The last step is display the result.

## **4.4 The structure of the program:**

### **4.4.1 MVC design model**

I apply MVC design pattern, MVC stands for the Model-View-Controller design pattern. MVC is a very popular design pattern aiming at divide the data model and view into two separate parts. In this way, the same data can be visualized in different ways. For example the same course information can appear the high level view and also it can appear in the course level view. Controller is used to mapping the data to the view that presented to the user.

The advantage that we can get from applying the MVC model is obvious.

First, it gives us the Low coupling feature which separates the view layer and the model layer wherefore change in the view don't affect the model layer. If the data processing method changes, we only need to change the model level and the other level remains the same.

Second, the MVC model makes the code easy to maintain.

### **4.4.2 The architecture of the program**

I divide the how structure into 5 modules:

The first module is in charge of the data procurement from the data base and then store this information retrieved from data base into the pre-defined classed and load them into the Main memory.

The second part is to generate the nodes in the scene and combine the bunch of nodes together from the nodes generated above.

The third part is to generate the relationships among the nodes.

The forth part is the utility which mainly handle the arranging of the nodes, and the selection of the node and scene.

The last part is the Visualization module which visualize the scene generated above

The Sequence Diagram of the model can be show in Figure4.7

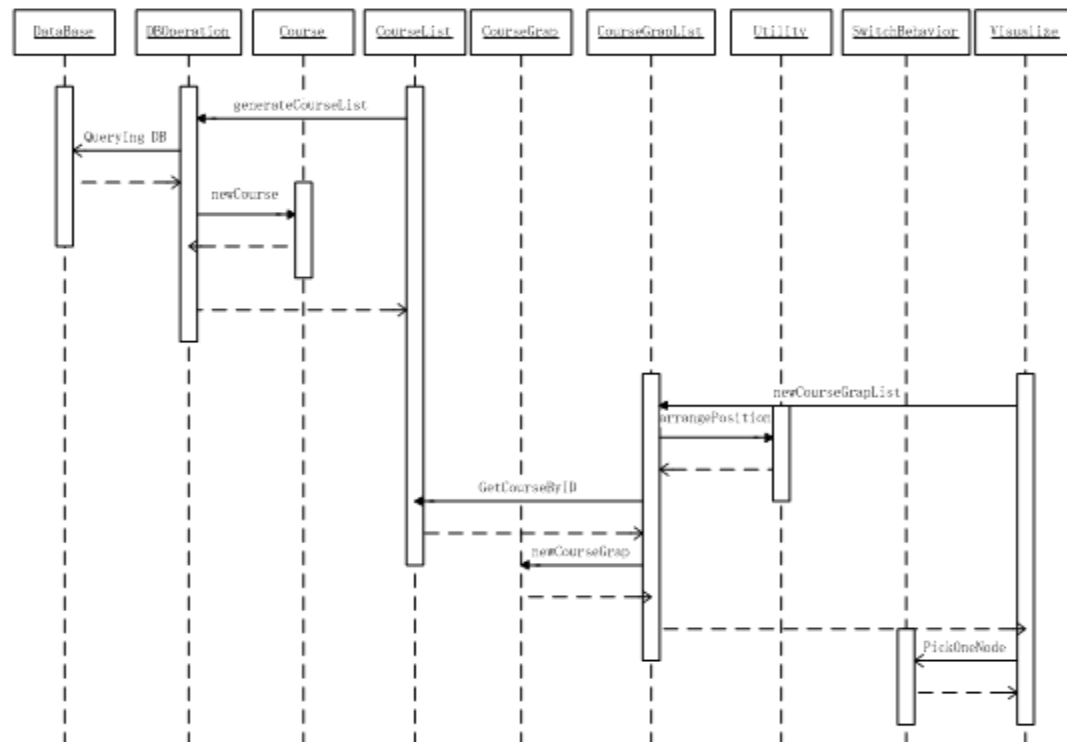


Figure 4.7 Sequence Diagram

The Sequence Diagram showed in the Figure4.7 is just one example based on the course information, yet actually there is a lot of information that is interchanged between the modular.

### 4.4.3 The design of the classes

In the package “com.anu.edu.au.database”, there exist 6 kinds of classes:

1. DBConnection: which is use to establish connection with the database.
2. DBOperation: which is used to retrieve information and generate entities list for each entities in the package.
3. Entity: is a single node that stores all the information about a specific entity.
4. EntityList: is a list which contains a group of entities.
5. EntityRelationship: is used to store the relationship information between entities.

6. EntityRelationShiplist: contains the same kind of EntityRelationship generated above.

In the Package “com.anu.edu.au.entities”, there exist 7 kinds of classes:

1. Entities: this class is the father type of the entire Entities graph class. This class contains the basic attributes which every Entities graph will have. The main attributes in the class is the ambient color, diffuse color, emissive color, specular color, and shininess of the entities demonstrated in the graph. Also I give the coordinate of the name text for each entity. I give each of the attribute an initial value except the coordinate for the text, because the position of the text should be arranged based on the entity itself.
2. Line: This is the super class for all the EntitiesRelationshipGraps, which predefined the color of the line.
3. EntityGrap: this class will extend the Entities which mean each EntityGrap class will inherit all the attribute and function of the father class. And this class adds an Entity which contains the information to visualize and a relative coordinates to the group of itself.

By using the generateXXX(), we can arrange the Transform Group which is a container including all the node to be displayed. This function is the core function that generates the atom node, and we will use the transform group as a whole.

The detail of the function is shown below in the figure 4.8

```
public TransformGroup generateCourse() {
    Box courseGrap=new Box(0.2f,0.2f,0.2f,super.app);
    courseGrap.setUserData("courseID "+course.getCourseID()+"/n"
        +"description:"+course.getDescription());
    FontExtrusion extrusion=new FontExtrusion();
    Font3D f3d=new Font3D(new Font("",Font.PLAIN,1),extrusion);
    Text3D courseText=new Text3D(f3d,course.getCourseName(),textCoord);
    courseText.setAlignment(Text3D.ALIGN_CENTER);
    Shape3D sh=new Shape3D();
    sh.setGeometry(courseText);
    sh.setAppearance(app);
    Transform3D textXfm=new Transform3D();
    textXfm.setScale(0.15);
    TransformGroup textXfmGrp=new TransformGroup(textXfm);
    textXfmGrp.addChild(sh);
    Transform3D courseXfm=new Transform3D();
    courseXfm.set(createPosition());
    TransformGroup courseXfmGrp=new TransformGroup(courseXfm);
    courseXfmGrp.addChild(courseGrap);
    courseXfmGrp.addChild(textXfmGrp);
    return courseXfmGrp;
}
```

Figure 4.8 the detailed function for generating a Transform Group.

The first line is used to generate a Box shape representing the course node, then we append some extra data to it which stands for the detailed information of this course. After that we set the font of the Text that will be displayed underneath. I set the font size to 1 which is the smallest in the library however it is still too large for the application as it shut out of the box shape. This is a troublesome problem new; nevertheless, I find a solution to this puzzle. I apply a scale to the text and by scaling it to 15% of its original size, it looks well. You can compare the unscaled shape in Figure 4.9 with the scaled one in Figure 4.10.



Figure 4.9 The unscaled graph

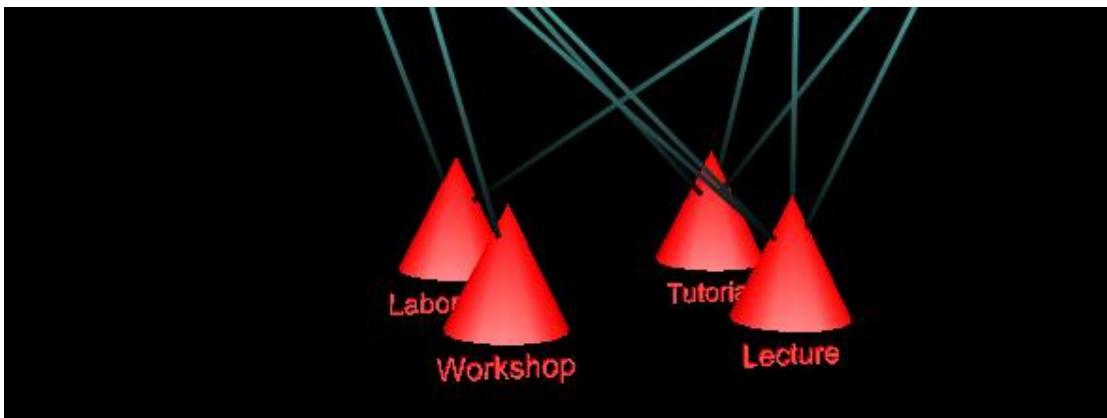


Figure 4.10 The scaled graph

Then I combine the text and course Transform group together to a new transform group. By doing this we can always use the new transform group as a whole and don't have to arrange the size and position of the text and the shape which representing the course. Because both the coordinate and the scale in the transform group is relative coordinate and relative scale, which will change according to the reference coordinate system. This design makes the transform group in another word node reusable.

#### 4. EntityGrapList

This is a container which contains a group of node which generated by the above class. This class contains the attributes relative coordinates, a list of EntityGraps and the list of course that is ready to be displayed. There are two core functions in the class, generateCourseGrapList() and generateCourseList(). The first is used to generate the CourseGrapList() from the courseList offered in the attribute and the second function is used to convert and integrate the course graph in the CourseGrapList into a single graph. The detailed code about these two functions can be found in Figure 4.11.

And the Utility.arrangePosition(numOfCourse, 2.0f) function is used to arrange the position of the nodes in this transform group automatically according to the number of objectives in the list. And I wrote several algorithms to allocate the proper coordinates to each node.

```
public void generateCourseGrapList() {
    int numOfCourse=courseList.getCourseList().size();
    float[][] positionList=Utility.arrangePosition(numOfCourse, 2.0f);
    for(int i=0;i<numOfCourse;i++){
        CourseGrap courseGrap=new CourseGrap(
            new Point3f(0.0f,-2.5f,0.0f),courseList.getCourseList().get(i),positionList[i]);
        courseGrapList.add(courseGrap);
    }
}

public TransformGroup generateCourseList() {
    Transform3D courseXfm=new Transform3D();
    courseXfm.set(creatPosition());
    TransformGroup courseXfmGrp=new TransformGroup(courseXfm);
    for(int i=0;i<courseGrapList.size();i++){
        courseXfmGrp.addChild(courseGrapList.get(i).generateCourse());
    }
    return courseXfmGrp;
}
```

Figure 4.11 The generateCourseGrapList and generateCourseList

And the result of this class can be found in Figure 4.12

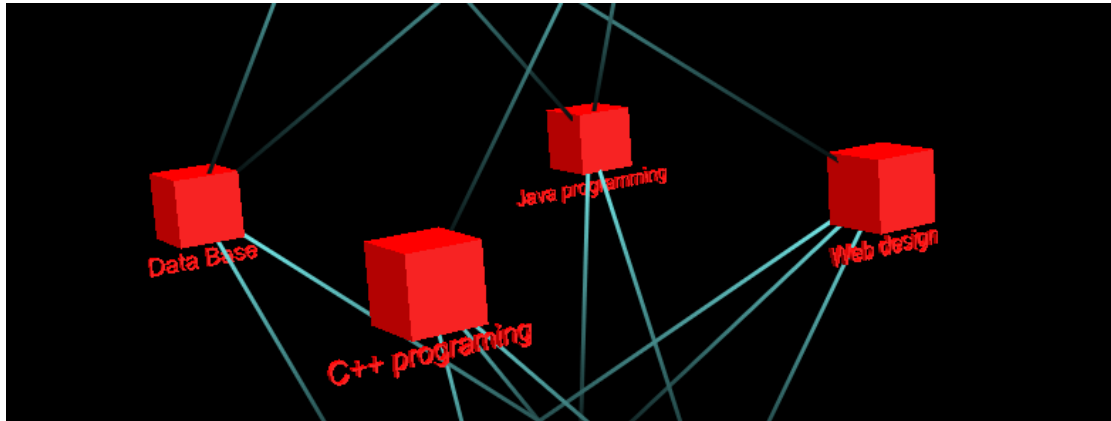


Figure 4.12 The effect of CourseGrapList

#### 5. EntitiesRelationshipGrap

This class is used to generate one line linking two entities, which represents there exists a relationship between the two entities. This class contains two EntityGrap which is actually the two entities that have a certain relationship. And in this class we generate a 3D line by using the coordinates of the two entities as two endpoints of the line, and set width of it.

#### 6. EntitiesRelationshipGrapList

This class contains the group of lines that representing the relationships among the Entities in the two entitiesList passed into this class as attributes. The procedure is similar to the EntityGrapList

#### 7. Utility

The Utilities class contains the algorithm used to allocating relative coordinates for the entities list automatically.

The detailed code of it is showed below in the Figure 4.13

```

public static float[][] arrangePosition(int numberOfEntity, float radius) {
    float[][] position = new float[numberOfEntity][3];
    double angle = Math.PI / numberOfEntity;
    for (int i = 0; i < numberOfEntity; i++) {
        float x = 0;
        float y = 0;
        float z = 0;
        if (angle >= 0 && angle <= (Math.PI / 2)) {
            x = (float) (radius * Math.sin(angle));
            y = 0.0f;
            z = (float) (radius * Math.cos(angle));
        }
        else if (angle > (Math.PI / 2) && angle <= (Math.PI)) {
            x = (float) (radius * Math.cos(angle - (Math.PI / 2)));
            y = 0.0f;
            z = - (float) (radius * Math.sin(angle - (Math.PI / 2)));
        }
        else if (angle > Math.PI && angle <= (1.5 * Math.PI)) {
            x = - (float) (radius * Math.sin(angle - (Math.PI)));
            y = 0.0f;
            z = - (float) (radius * Math.cos(angle - (Math.PI)));
        }
        else if (angle > (1.5 * Math.PI) && angle < 2 * Math.PI) {
            x = - (float) (radius * Math.cos(angle - (1.5 * Math.PI)));
            y = 0.0f;
            z = (float) (radius * Math.sin(angle - (1.5 * Math.PI)));
        }
        position[i][0] = x;
        position[i][1] = y;
        position[i][2] = z;
        angle += 2 * Math.PI / numberOfEntity;
    }
}

```

Figure 4.13 Arrange the nodes in a polygon

The result of arrange 3, 4, 5 nodes is in figure 4.14

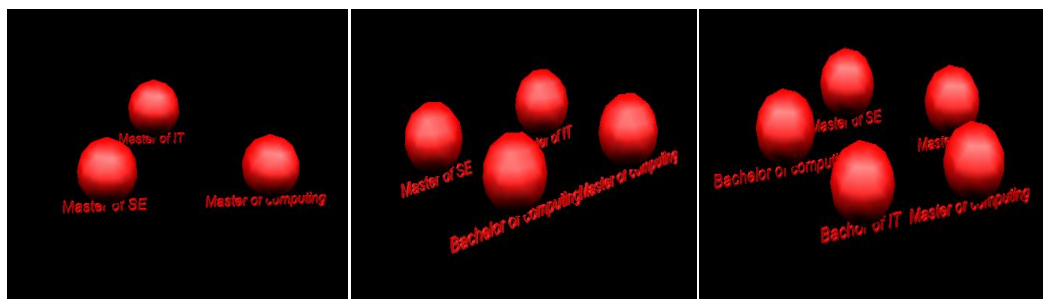


Figure 4.14 The allocation of 3, 4, 5 nodes

In the Package “com.anu.edu.au.view”, there exist 3 kinds of classes:

#### 1. Visualization:

This is the class that implements the Visualizations. In the buildContentBranch function I implement rotating, zoom in, zoom out and translation on the graph with a mouse. Then I generate the several scenes in the function named buildshap and set the initial scene to the top level overview layer. Furthermore I generated the top



and second layers of the graph. Also, in the addLight function I add two directional lights to the scent which will make it more real.

## 2. Switch Behavior

In this class, I implement detection of mouse picking a node. If you pick some node with your mouse, the text panel at the top of the application will give you the detailed information of the node picked. Then switch the scene to the detailed view of the node choose.

## 3. Utility

This class is used to determine which scene the node being picked is belonging to. The detailed code is shown below in Figure 4.15

```
public static int calculateSwitch(String userData, int programNum, int courseNum) {  
    int s=0;  
    StringTokenizer lineTokenizer=new StringTokenizer(userData, " ");  
    String nameOfEntity=lineTokenizer.nextToken();  
    int id=Integer.parseInt(lineTokenizer.nextToken());  
    if(nameOfEntity.equals("programID")) {  
        s=id;  
    }  
    else if(nameOfEntity.equals("courseID")) {  
        s=programNum+id;  
    }  
    else if(nameOfEntity.equals("deliveryMethodID")) {  
        s=programNum+courseNum+id;  
    }  
    return s;  
}
```

Figure 4.15 Switch scene function

The function reads the usetData that put into the shapes before, split the user data into several and take out the Entitiy ID. Then use this ID to deciede whether this Entity has a detailed view to show. If it has, then change the scene, if not, just return to the higher level or remain unchanged.

## 4.5 The result of the Visualization

The result of the Visualization is based on some test data from the database.

The high level visualization is shown in Figure 4.16 and the Program Level visualization is shown in figure 4.17

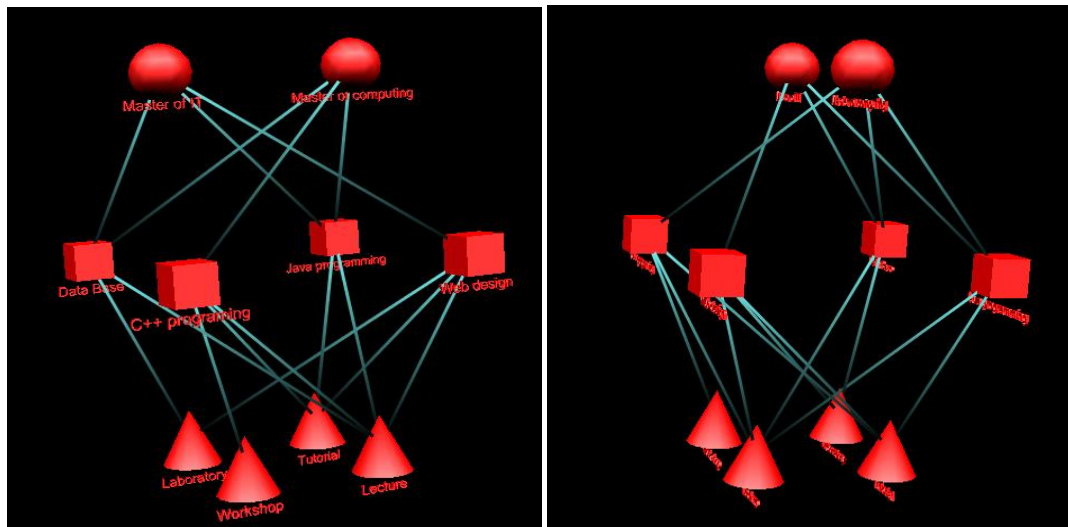


Figure 4.16 High level visualization

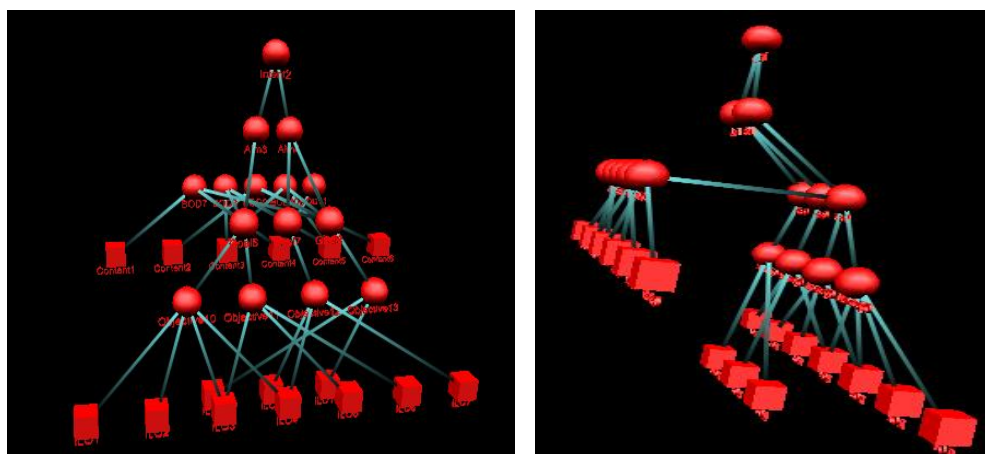


Figure 4.17 Program Level visualization

## 4.6 Test

I apply a verification test on the application.

First, get the information from the data base.

Second, run the program and compare the result with the original information from the data base to see if there is some information visualized is uncorrected, or missed.

Finally, record the result for future use.

I apply the three steps on each of the information in the data base.

Result:

All the information can be visualization correctly.

Information in the data base is shown in the table 4.1:

ProgramID	ProgramName	Description
1	Master of computing	Advanced computer science
2	Master of IT	Basic information technology

CourseID	CourseName	Description
1	Web design	Basic Web design knowledge
2	Java programming	Basic Java language
3	Data Base	mysql and Sql language
4	C++ programing	C++ language

DeliveryMethodID	DeliveryMethodName	Description
1	Lecture	Give lectures
2	Tutorial	Give tutorials
3	Laboratory	Arrange labs
4	Workshop	Arrange Workshops

ProgramCourseRID	ProgramID	CourseID	Description
1	2	1	Null
2	2	2	Null
3	2	3	Null
4	1	2	Null
5	1	3	Null
6	1	3	Null

CourseDeliveryMethodRID	CourseID	DeliveryMethodID	Description
1	1	1	Null
2	1	2	Null
3	1	3	Null
4	2	1	Null
5	2	2	Null
6	3	1	Null
7	3	3	Null
8	4	1	Null
9	4	2	Null
10	4	4	Null

Table 4.1 Information from the data base

Step 2: Run the program, and the result is shown below in the figure 4.18

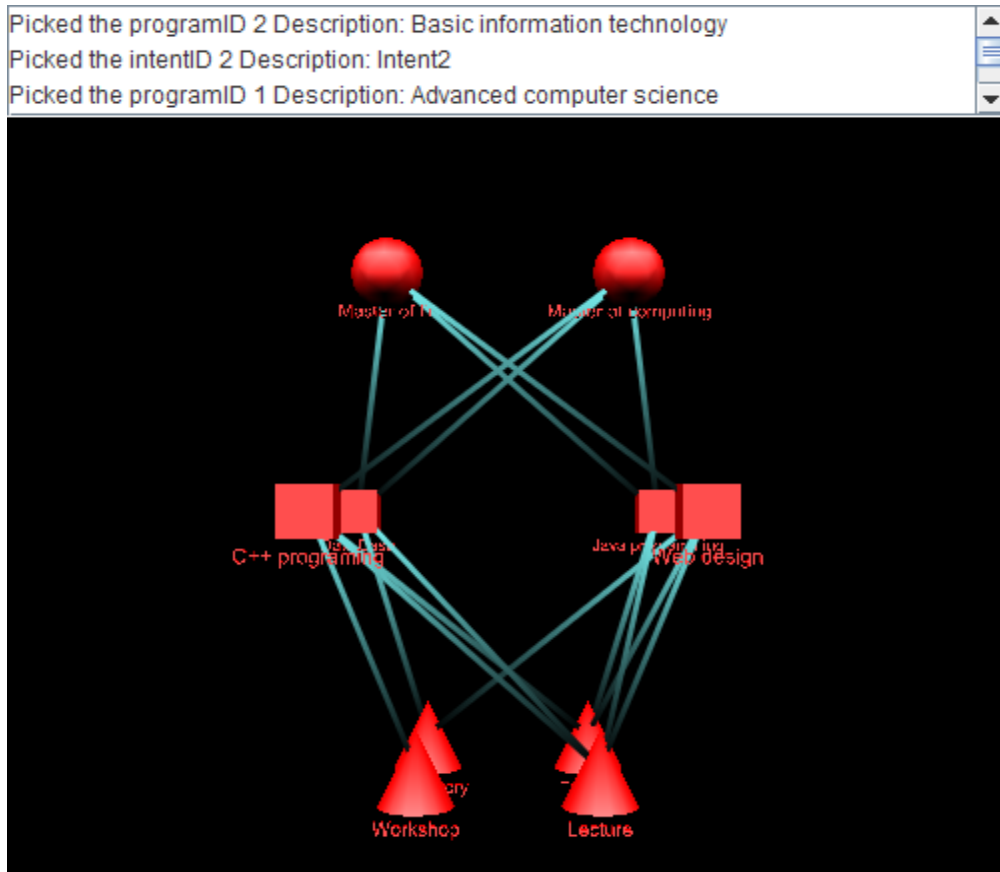


Figure 4.18 Result of the information above

All of the Programs, courses and Delivery Methods are shown in the graph with the correct name. Furthermore the relationship between programs and courses is generated correctly as well as the relationships between courses and delivery methods.

And Then I test the program level view and the course level view, all the information is covered correctly in the scene graph.

Then I add more information into the data base to see whether the application will work out the correct result.

Also I follow the steps above, the result showing that when the objective in the program view is large than 14, the information is correct illustrated, however, the structure of the graph become complex, and the text in the

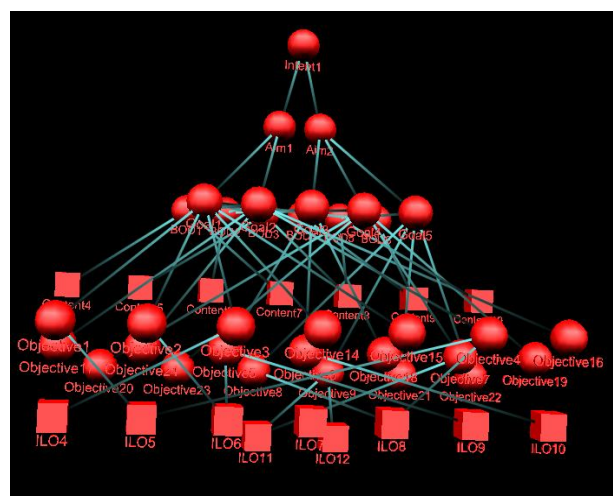


Figure 4.19 P-level visualization

inner layer is not easy to recognized. The result of a program Level view is shown in figure 4.19

You can see the graph still works, however, some entities is hide from the front entities which is not readable, so the effect of the visualization is not perfect.

This problem also exists in the other layers.

## 4.7 Problems and solutions

1. The first problem is that I can put the text on the objects.

The sphere object in the Java 3D was generated as the primitive type and you can attach user data on the primitive however the user data cannot be displayed. Another way to attach the text on the primitive is by defining the textures; however the generated texture is a picture than cannot be generated dynamically. So, I just attach the text under the objects and combine them together. In this way I can display some text that can explain what the object is.

2. The second problem is also related to the text. The text rotates together with the object when we use the mouse to rotate the graph.

We hope that when we rotate the graph the text is not rotated and always facing us so we can always read it.

This problem can be fixed by using the billboard behavior, however, by applying this behavior to the text, its position also changed. So, I should do more research on this behavior.

3. The third problem is that I can't change the scale and drag it to different place to make it clearer.

This problem may be solving by using the Animation features in the Java3D. Due to there exist a lot of Local coordinate systems, so to get the position and calculate the position after drag is not as easy as they are in the 2D space.

4. When I store more information into the data base which means the relationships among the curriculum is complex, it takes a long time to generate the scene. When there are 2 programs in the data base, it takes around 5 seconds to generate the scene graphs. May be by change the algorithm of allocating position or changing the structure I store the information, the program may cost less time.

5. The outer object will block out the inner ones.

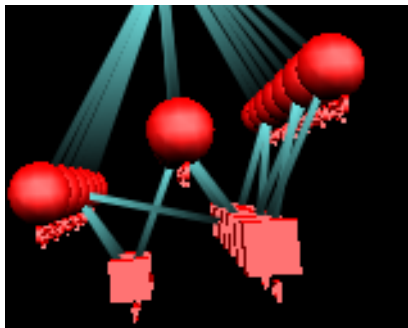
I arrange the object in several layers, and make the position of inner layer a little

lower than the outer layer. In this way the inner layer will not be blocked.

## 4.8 Analysis of the result

The good feature of the application is:

- The application uses a pretty simple way to reveal the relationship among the curriculums. When there is a relationship between two entities, I link the two entities with a line. This is easy to understand, and it is very intuitionistic.
- Divide the whole program into three levels, the high overview level, the program level, the course level. It is easy for the user to understand the structure of the whole program step by step.
- You can interact with the graph, by clicking on an object; it will give you a detailed explanation of the object on the text area above. And If the object have a lower lever view, the program will demonstrate the more detailed view instead.
- Once the scene is generated you don't have to wait for any time when changing the view, because it is already generated.
- The application wills automatically allocated position for the entities according



to the quantities of them. And if there is no more space at front, the application will make good use of the spatial space, and arrange the other object inside. And the inner object will be a bit lower than the outer ones. So, the inner ones will not be blocked out by the outer ones as shown in figure 4.20.

Figure 4.20

- You can rotate, zoom in, zoom out, and translate the graph, which will help you see every aspect of the graph.

The bad thing about the design

- When you rotate the graph, you cannot see the text very clear, because the text is not facing you.
- You cannot change the scale and drag the object to other places.
- When there are a lot of entities in the graph, it sometimes hard to read the text

and even cannot figure out which two objects have a line between them.

- You cannot edit, delete add a node, link on the graph.
- The object in the graph is tedious and lacking variation

## 4.9 Reflection

I have learned a lot of Java3D technology, time allocation in a project and the importance of research on existing similar software or previous research.

If I have the chance to start the project all over again, I will do more research on the previous work done on this area instead of hurried to start doing the project by myself. Research on the previous job will save you a lot of time, and it will help you avoid detours. Also I will apply the Rapid Prototype Model, because this is a project that doesn't have a very clear requirement at beginning and by applying this software development model, I can decrease the risk brought by the indefinite requirements.

In the techniques level I found that though Java3D is a good 3D developing language, however it seems that in the visualization area it doesn't work very well. This project requires a lot of interaction with the objects in the graph, however the Java3D doesn't provide enough action listener handling the interaction, and the method to select a specific object is not very good compared with VRML language.

Then Java 3D mainly focus on creating animations and games. So if I could do it again, I will engage more time on selecting a more suitable technology to implement the project.

## 5. Future work

- Add the text on the object itself instead of appending it under the object.
- Using 3DS Max to create the object model and imported them into Java 3D
- Implement the scale and drag function which allows the user to drag the object to any where they want.
- Improve the data structure in Data base to reduce the time used to get information.
- Simplify the code, use more inherit and interface to increase the reuse of the code.
- Add the function allowing user to add, delete, and rewrite the node or link.

- Add the function which will give a simple feedback according to the operation taken by the user. For example the system will give a prompt that you cannot delete the node because the ILOs is not meet.

## 6. Conclusion

In this report, I first give objective of this project then I introduce the background of 3D visualization and the Constructive Alignment theory. After that I give the design detailed of the program. Finally I run a simple verification test on the program and make some verification about the result.

In this project, I have the following result.

1. The 3D visualization is a good way to demonstrate the relationships among the curriculums.
2. The A-content Web service API is not support the information retrieving at a very detailed level, and only the basic information about the course can be exported. Furthermore the A-content don't have a special function that stores the relationships among the curriculums and all we can do is to store the relationships in a simple format file.
3. The Data model of the data base in this report is suitable for visualizing the relationship, because the result shows no errors by using this model.
4. The objective used in the program to show the information and the relationships is very simple and understandable.
5. The dividing of the whole curriculum into the high level view, program view and course level view make it more comprehensive for user to understand.
6. The program can generated the 3D relationship graph according to the information stored in the data base automatically.
7. The method used in this project dealing with allocating position is effective when the quantity of the curriculum is not so huge.
8. The Java3D technology is not so powerful when dealing with the interaction with the objectives in the 3D graph.

To sum up, there is no much research or work done previously on the 3D visualization of the curriculum information, so all of the design is done by me. So there are some design defects in this program, such as the time using to generating the scene graph is a little long, the interaction between the user and the graph is not



enough, the text and relationships is unreadable when the quantity of the object in the graph is too much. However, the basic objective is reached, this project find a feasible way to present the information of the curriculum in a 3D graph.

This project is just a start on researching on the 3D visualization of the curriculum, there still a long way to go to find some better solution to this problem.

## 7. References

- [1] Biggs, J. *Aligning the curriculum to promote good learning. Constructive Alignment in Action*, 2002.
- [2] Marton, F. Booth, S. *Learning and Awareness*. Lawrence Erlbaum, Mahwah (1997)
- [3] Referenced on 17<sup>th</sup>, April, 2011 from the web site “Higher Education Academy Engineering Subject Center” and the URL address is listed below:  
<http://www.engsc.ac.uk/learning-and-teaching-theory-guide/constructive-alignment>
- [4] Claus Brabrand, *Constructive Alignment for Teaching Model-Based Design for Concurrency*, 2008
- [5] B.H. McCormick. A. Defanit, and M.D. Brown. “*Visualization in scientific Computing*” report of the NFS Advisory Panel on Graphics, Image Processing and workstations, 1987
- [6] L. Rosenblum et al. *Scientific Visualization Advances and Challenges*.
- [7] Michael Friendly, *Milestones in the history of thematic cartography, statistical graphics, and data visualization*.
- [8] Referenced on 26<sup>th</sup>, April, 2011, from the website AContent with the URL:  
[http://atutor.ca/acontent/demo/documentation/web\\_service\\_api.php](http://atutor.ca/acontent/demo/documentation/web_service_api.php)
- [9] Lynette Johns-Boast, unpublished Model of curriculum.

## 8. Appendix

Class Diagram is shown in the Figure 8.1, 8.2 and 8.3

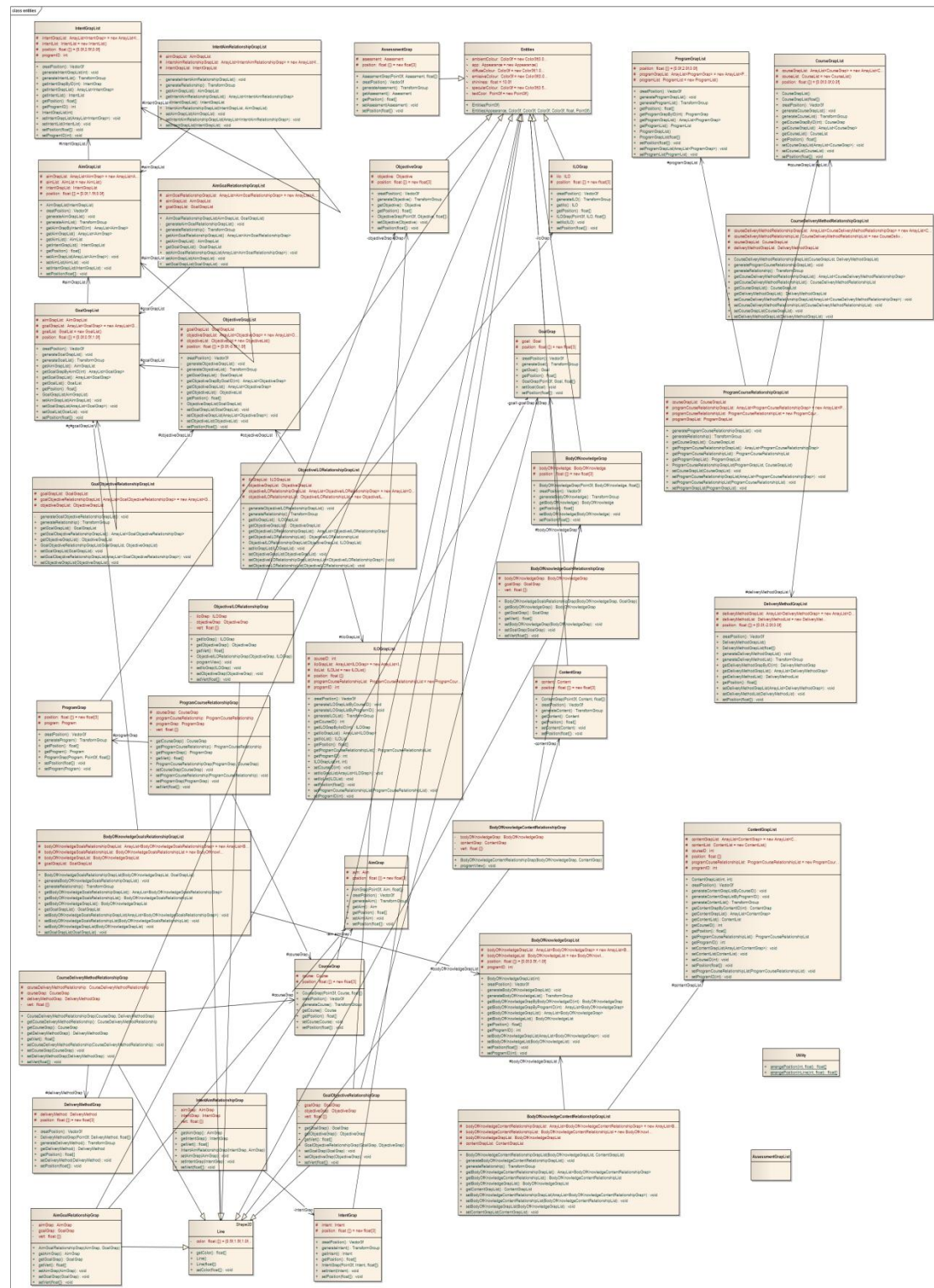


Figure 8.1 Class diagram of the package Entities

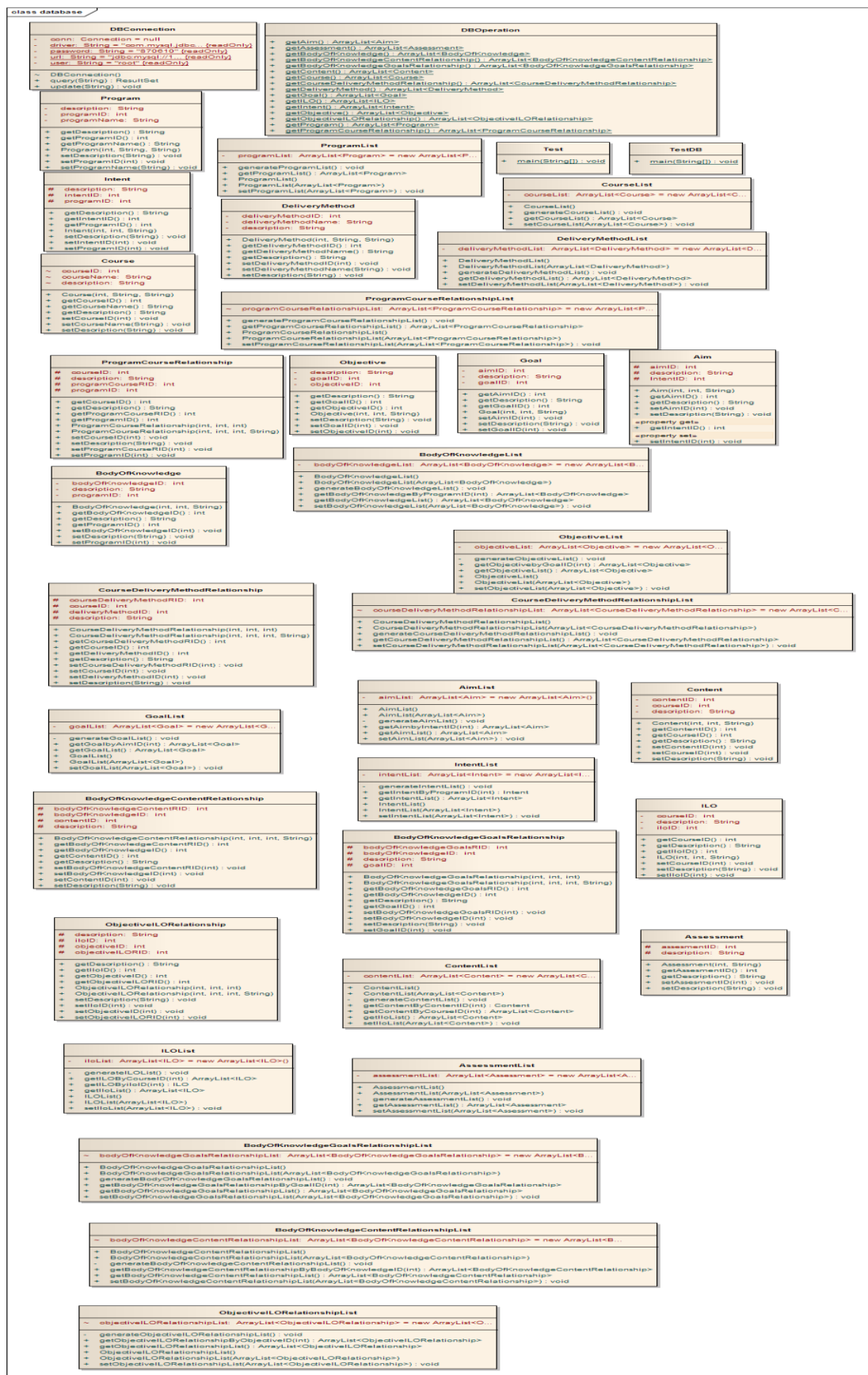


Figure 8.2 Class diagram of the package DataBase

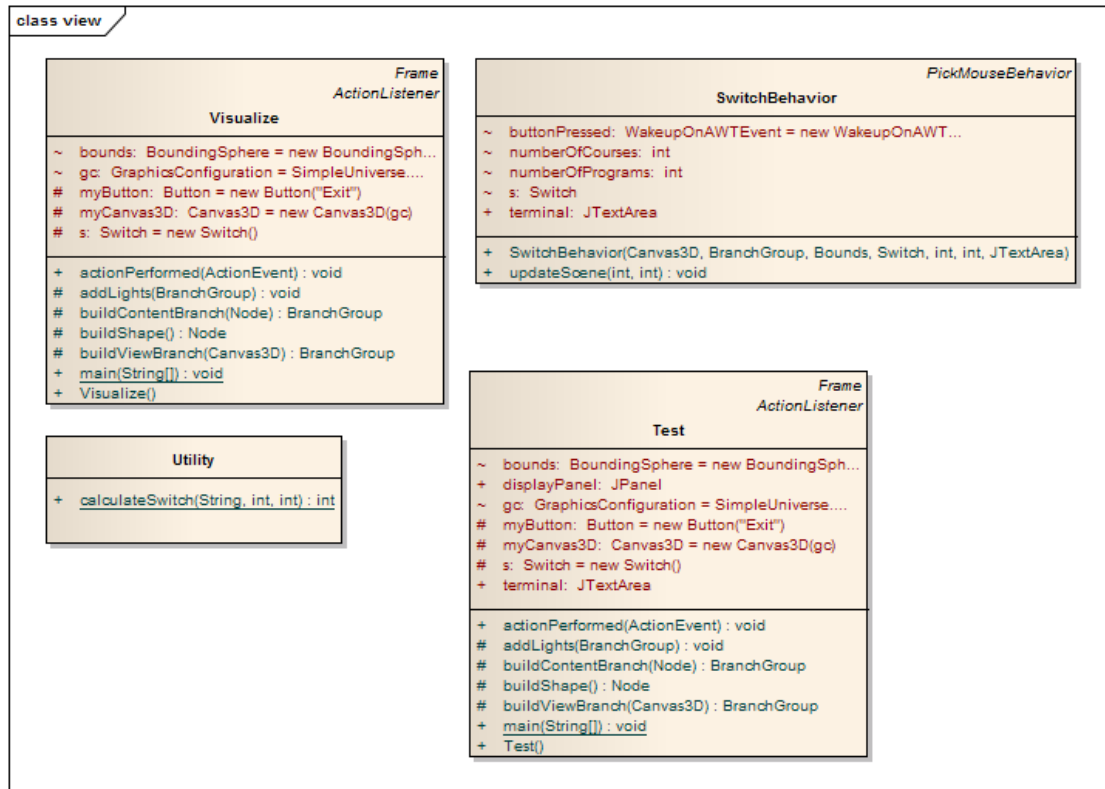


Figure 8.3 Class diagram of the package Visualization

Data base design Scheme is shown in figure 8.4

User's manual

1. Establish a database using the data base design scheme.
2. Run the program, and a high level over view of the curriculum information will appear.
3. Click on the object and if there is a more detailed graph, the program will shown the low level graph, and in the text area above, the detailed information about this object will appear.
4. If you update the information In the database the graph will

auto generated when you run the program next time.

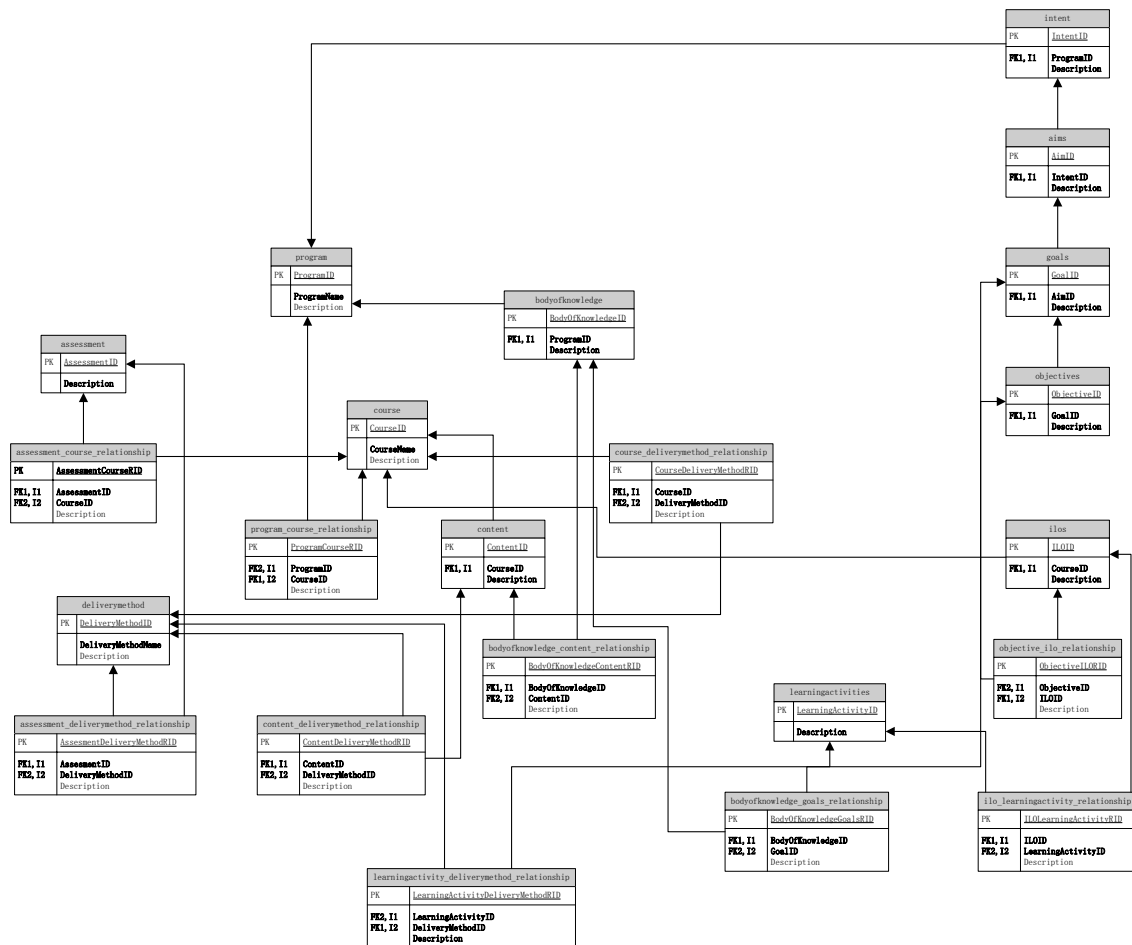


Figure 8.4 Data base design scheme